

TITLE OF THE INVENTION
METHOD, COMPUTER PROGRAM PRODUCT AND SYSTEM
FOR PROVIDING INFORMATION PROCESSING SERVICE

5 CROSS-REFERENCE TO RELATED APPLICATIONS

This application is based upon and claims the benefit of priority from the prior Japanese Patent Applications No. 2001-079353, filed March 19, 2001; and No. 2002-069188, filed March 13, 2002, the entire contents of which are incorporated herein by reference.

10 BACKGROUND OF THE INVENTION

1. Field of the Invention

本発明は、情報処理サービス提供者を支援する情報処理サービス提供方法及びコンピュータプログラムプロダクト並びにシステムに関する。

2. Description of the Related Art

15 サービス提供者とは、例えばビジネスに使用される業務ソフトウェアなどのアプリケーション・ソフトウェア（以下、アプリケーションという）の機能、又はその機能を提供する事業者である。サービス提供者は、アプリケーションによって得られた結果を、ユーザ又はシステムなどの依頼側に出力する。

A S P (Application Service Provider) は、サービス提供者の一例である。

20 A S P は、例えば統合業務パッケージなどの大規模な業務システムの機能をレンタルの対象とする。

また、A S P は、文書編集ソフトウェア、表計算ソフトウェアなどのような日常的に使用されるアプリケーションもレンタルの対象とする。

ユーザは、自己の端末にアプリケーションをインストールしなくとも、A S P

のレンタルアプリケーションを使用し、作業可能である。

これにより、ユーザ側では、企業の情報システム部門にとって負担となっていたアプリケーションのインストール、管理、アップグレードなどの費用、労力を軽減できる。

- 5 一般的に、ユーザは、Webブラウザを使用してASPにアクセスする。しかしながら、Webブラウザを使用することなくASPにアクセスする場合もある。例えば、ユーザは、携帯電話専用のブラウザを使用してASPにアクセスする。また、ユーザは、ASPから期限付きでレンタルされるソフトウェアを自己のハードディスクにダウンロードし、使用する。

10 サービス提供者は、サービス料を得るために、ユーザの使用量に応じて課金する機能及びユーザ情報の管理機能を具備する必要がある。したがって、サービス提供者は、ユーザに使用されるアプリケーションと連携動作する課金管理機能及びユーザ管理機能を開発し、運用する必要がある。

15 サービス提供者は、サーバのダウンや異なるユーザ間での情報漏洩を防止する必要がある。そのため、一般的に、サービス提供者は、サービスを提供する目的に特化した施設及び運用形態を提供するデータセンタ又はインターネットデータセンタ（以下、iDCという）にサーバを設置し、サーバダウンや情報漏洩を防止する。

20 データセンタ又はiDCは、物理的なインフラ及び運用管理をサービス提供者に提供する。

しかしながら、サービス提供者がサービスを提供するために必要な機能の中には、データセンタ又はiDCを使用するのみでは実現されない機能がある。

25 例えば、データセンタ又はiDCを使用するのみでは、アプリケーションの使用量に応じた課金管理の機能、請求書作成支援機能、請求書発送を支援する機能、ユーザからの問い合わせに対して返答する機能等が十分に実現されない。

サービス提供者の中には、サービスを提供するために必要な全ての機能を自己で開発又は運用する技術力を持たない者もいる。このように、技術力が十分でないサービス提供者が上記機能を実行するサービスを提供するのは困難であり、サービス提供の障害となる。

BRIEF SUMMARY OF THE INVENTION

本発明は、サービス提供者を支援するための情報処理サービス提供方法及びコンピュータプログラムプロダクト並びに情報処理サービス提供システムを提供することを目的とする。

5 According to an embodiment of the present invention, there is provided a method for providing information processing service by a computer, comprising

ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出し、

前記イベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照し、

前記ライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still another embodiment of the present invention, there is provided a method for providing information processing service by a computer, comprising

ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録し、

前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に、記録された前記運用状態を示すデータを提供する。

According to a still another embodiment of the present invention, there is provided a method for providing information processing service by a computer, comprising

ネットワークを経由して提供される情報処理サービスのリクエストを取得し、

前記リクエストに応じて、前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still another embodiment of the present invention, there is provided a method for providing information processing service by a computer, comprising

ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する付加的な情報処理サービスのリクエストを取得し、

前記リクエストに応じて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

5 According to a still another embodiment of the present invention, there is provided an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein, the computer program code means comprising

10 a detection computer readable program code that ネットワークを経由して提供される情報処理サービスに対する付加的な情報処理サービスを依頼するイベントを検出する，

a reference computer readable program code that 前記イベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照する， and

15 a request computer readable program code that 前記ライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still another embodiment of the present invention, there is provided an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein, the computer program code means comprising

20 a record computer readable program code that ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録する， and

25 a request computer readable program code that 前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に、記録された前記運用状態を示すデータを提供する。

According to a still another embodiment of the present invention, there is provided an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein, the

computer program code means comprising

a acquisition computer readable program code that ネットワークを経由して提供される情報処理サービスのリクエストを取得する, and

a request computer readable program code that 前記リクエストに応じて、

- 5 前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still another embodiment of the present invention, there is provided an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein, the
10 computer program code means comprising

a acquisition computer readable program code that ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する付加的な情報処理サービスのリクエストを取得する, and

a request computer readable program code that 前記リクエストに応じて、
15 前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still another embodiment of the present invention, there is provided a system for providing information processing service comprising

a detection unit that ネットワークを経由して提供される情報処理サービス
20 に対する付加的な情報処理サービスを依頼するイベントを検出する,

a reference unit that 前記イベントが検出された場合に、前記付加的な情報処理サービスを提供する付加的機能の呼び出しを定義したライブラリを参照する,
and

a request unit that 前記ライブラリの参照結果に基づいて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。
25

According to a still another embodiment of the present invention, there is provided a system for providing information processing service comprising

a record unit that ネットワークを経由して提供される情報処理サービスの運用状態を示すデータを記録する, and

a request unit that 前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に、記録された前記運用状態を示すデータを提供する。

5 According to a still another embodiment of the present invention, there is provided a system for providing information processing service comprising

a acquisition unit that ネットワークを経由して提供される情報処理サービスのリクエストを取得する, and

10 a request unit 前記リクエストに応じて、前記情報処理サービスに対して付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

According to a still further embodiment of the present invention, there is provided a system for providing information processing service comprising

15 a acquisition unit that ネットワークを経由して提供される情報処理サービスの処理に基づいて発生する付加的な情報処理サービスのリクエストを取得する, and

a request unit that 前記リクエストに応じて、前記付加的な情報処理サービスを提供する付加的機能に実行を依頼する。

20 Additional objects and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out
25 hereinbefore.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the

present invention and, together with the general description given above and the detailed description of the embodiments given below, serve to explain the principles of the present invention in which:

図1は、本発明の第1の実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図；

図2は、同実施の形態に係る情報処理サービス提供システムの構成の変形例を示すブロック図；

図3は、本発明の第2の実施の形態に係る情報処理サービス提供システムの連携状態の一例を示すブロック図；

図4は、同実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図；

図5は、ユーザ側リクエスト取得機能による連携状態の一例を示すブロック図；

図6は、アプリケーション側リクエスト取得機能による連携状態の一例を示すブロック図；

図7は、リクエスト取得機能を含む情報処理サービス提供システムのアーキテクチャの一例を示すブロック図；

図8は、同実施の形態に係る情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係の一例を示すブロック図；

図9は、リクエスト取得機能の構造の一例を示すクラス図；

図10は、リクエスト取得機能の構造と具体的なサービスとの関係の一例を示すクラス図；

図11は、ユーザ側リクエスト取得機能の前段動作の一例を示すシーケンス図；

図12は、ユーザ側リクエスト取得機能の後段動作の一例を示すシーケンス図；

図13は、アプリケーション側リクエスト取得機能の前段動作の一例を示すシーケンス図；

図14は、アプリケーション側リクエスト取得機能の後段動作の一例を示すシ

一ケンス図；

図15は、本発明の第3の実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図。

DETAILED DESCRIPTION OF THE INVENTION

5 以下、図面を参照しながら本発明の実施の形態について説明する。なお、以下に示す各図において、同一の部分については同一の符号を付する。

以下の説明においては、主にユーザが例えばWebブラウザなどを操作してリクエストを発信する場合について説明する。しかしながら、これに限定されるものではなく、プログラム、コンピュータ、コンピュータシステム、オブジェクト、モジュール、プロセスなどがリクエストを発信してもよい。

(第1の実施の形態)

本実施の形態においては、情報処理サービスに対する付加的な情報処理サービス（以下、「付加サービス」という）を提供する情報処理サービス提供システムについて説明する。

なお、本実施の形態においては、ASP業務を行うための情報技術サービスが情報処理サービスである場合を例として説明する。ASPは、サービスの依頼側にアプリケーションサービスを提供する。付加サービスの一例としては、ASPに対する運用管理サービスがある。

図1は、本実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図である。

情報処理サービス提供システムのユーザ（エンドユーザ及びユーザ企業など）2は、自己の操作するクライアント3からインターネット4を経由してASPサーバ5a又はASPサーバ5bにリクエストを提供する。このリクエストに基づいてASPサーバ5aのアプリケーション6a又はASPサーバ5bのアプリケーション6bは稼動する。その結果、ユーザ2は、ASPサービスを受ける。

ASPサーバ5a、5bのイベント検出機能7は、付加サービスを依頼するイベントの発生を検出する。

例えば、ASPサーバ5a、5bは、新規のユーザにアクセスされた場合、この新規ユーザを管理するために新規ユーザの登録を必要とする。この場合、イベ

ント検出機能 7 は、新規のユーザからのアクセスを検出する。その他にも、イベント検出機能 7 は、ユーザの使用に応じてそのユーザに対して課金するためのイベント、またASPサーバ 5 a、5 b の障害発生を示すイベントを検出する。

ライブラリ 8 は、付加サービス 11 a～11 c を提供する機能の呼び出しに使
5 用されるデータ又はプログラムを格納している。ライブラリ 8 は、ASPサーバ
5 a、5 b から付加サービスを提供する機能を呼び出すための API (Application Program Interface) を具備する。

なお、付加サービスを提供する機能は、例えばプログラム、プロセス、モジュ
10 ル、アプリケーション、コンピュータ、コンピュータシステムなどで実現され
る。また、付加サービスを提供する機能に対する実行の依頼方法としては、例え
ば、付加サービスを提供する機能にリクエストを出力する手法、付加サービスを
提供する機能にデータを提供する手法、付加サービスを提供する機能の処理を呼
び出す方法などがある。

ライブラリ 8 は、参照機能 9 を実現するハードウェアと同一のハードウェアに
15 具備されない場合もある。そのため、参照機能 9 は、ネットワーク経由でライブ
ラリ 8 を参照可能とする。

参照機能 9 は、イベント検出機能 7 でイベントが検出された場合にライブラリ
8 を参照する。

依頼機能 10 は、参照機能 9 によるライブラリ 8 の参照結果を使用し、イベ
20 ント検出機能 7 で検出されたイベントの種別に応じた機能を呼び出す。

付加サービス 11 a～11 c は、そのサービスを提供する機能に実行が依頼さ
れると実現される。

上記図 1 では、付加サーバ 12 a によってユーザ管理サービス 11 a 及び課金
25 管理サービス 11 b が提供される。また、付加サーバ 12 b によってセキュリテ
ィサービス 11 c が提供される。

ASPサーバ 5 a、5 b と付加サーバ 12 a、12 b とは、ネットワーク 4 a
を介して接続されている。

なお、本実施の形態において、イベント検出機能 7、参照機能 9、ライブラリ
8、依頼機能 10 は ASPサーバ 5 a、5 b 側に配置されているが、例えば付加

サーバ12a、12b側に配置されてもよい。付加サーバ12a、12b側にイベント検出機能7が配置された場合、このイベント検出機能7はネットワーク4aを経由してASPサーバ5a、5bで発生するイベントを検出する。

また、本実施の形態においては、ASPサーバ5a、5bと付加サーバ12a、
5bとが異なるハードウェア上で稼動する場合を例として説明している。しかしながら、ASPサーバ5a、5bと付加サーバ12a、12bは同一のハードウェア上で稼動してもよい。

図2は、本実施の形態に係る情報処理サービス提供システムの構成の変形例を示すブロック図である。

ASPサーバ5cの記録機能13は、ASPサーバ5cの運用状態データをデータベース14に記録する。運用状態データには、例えばユーザ2のアプリケーション6cの使用回数、使用時間、ASPサーバ5cの動作履歴情報などがある。

ASPサーバ5cの依頼機能15は、付加サービスを提供する機能に対してデータベース14に記録されている運用状態データを出力する。

付加サーバ12aは、運用状態データを入力すると、ユーザ管理サービス11a又は課金管理サービス11bを提供し、付加サーバ12bは、運用状態データを入力すると、セキュリティサービス11cを提供する。

以上説明した情報処理サービス提供システムにおいては、ASPのアプリケーション6a、6bと付加サービス11a～11cを提供する機能とが連携して動作する。

したがって、例えばiDC業者が各種付加サービスを提供すれば、このiDC業者は自己のデータセンタの付加価値を高めて事業を行うことができる。

また、ユーザ2が複数のASPサービスを受ける場合、使用される複数のASPが共通の付加サービスを使用することにより、このユーザ2に対するユーザ管理や課金管理を複数のASP間で統一化することができる。

したがって、ユーザ2は、複数のASPについてシングルサインオン環境を使用可能であり、複数のASPへの料金の支払いを統合できる。

また、ASPは、ASPアプリケーションとしての仕様で作成されていないアプリケーションに対して、ASPアプリケーションに必要な機能を付加する付加

サービスを受け、容易にASPアプリケーションとして使用することができる。
したがって、ASPは、各種アプリケーションを容易にレンタルすることができ、
自己の提供するサービスを充実させてユーザの使用を促進させることができる。

- また、ASPは、自己のみでは十分なサービスを提供できない場合であっても、
5 他のASPによって提供される付加サービスを使用し、サービスの価値を高める
ことができる。

また、ASPは、付加サービスを他のASPに依頼することにより、自己が管
理する機能を削減でき、管理業務を簡易化させることができる。

なお、本実施の形態に係る情報処理サービス提供システムの各種機能7～10、
10 13、15は、記録媒体16に記録されたプログラムをコンピュータが読み取る
ことで実現可能である。

(第2の実施の形態)

本実施の形態においては、ASPサービスと付加サービスとを連携させる情報
処理サービス提供システムについて詳細に説明する。

15 図3は、本実施の形態に係る情報処理サービス提供システムの連携状態の一例
を示すブロック図である。

ユーザ2は、ASP18のアプリケーションによるASPサービス18a～18gを使用する。

ASPポータル17は、ユーザ2とASP18とのマッチメイク、ASP18
20 に関するマーケティングを行う。さらに、ASPポータル17は、ASP間の連
携とそのための情報処理サービス、ユーザに対するASP導入のためのコンサル
ティング、BPR(Business Process Re-engineering)支援サービスを提供す
る。

ASP18は、ASPアプリケーションによるASPサービス18a～18g
25 をユーザ2に提供する。

付加サービス業者19は、複数のASPに対する共通の情報処理サービス、A
SPの運用管理サービスなどの付加サービス11a～11jを提供する。

付加サービスの一種である共通運用管理サービス11jは、付加サービス業者
19が他の付加サービス業者の提供する付加サービス又は他のASPの提供する

A S P サービスなどを受けるための管理を行う。

i D C 業者 2 0 は、物理インフラ及び運用管理サービスを提供する。また、 i D C 業者 2 0 は、ホスティング(サーバ貸し)、ハウジング(場所貸し)を行う。

5 A S P 1 8 によって提供されるA S P サービス 1 8 a ~ 1 8 g は、必要に応じて付加サービス業者 1 9 によって提供される付加サービス 1 1 a ~ 1 1 j 及び i D C 業者 2 0 の i D C 2 1 を使用する。

i D C 2 1 上には、付加サービス 1 1 a ~ 1 1 j を提供するためのアプリケーションプラットフォーム 2 2 が具備されている。

10 図 4 は、本実施の形態に係る情報処理サービス提供システムの構成の一例を示すブロック図である。

本実施の形態では、付加サービス 1 1 a ~ 1 1 j をその役割に基づいて、ベーシックサービス、コミュニケーションサービス、ユーティリティサービス、バックエンドサービスの 4 種に分類する。

15 ベーシックサービスは、A S P サービス 1 8 a ~ 1 8 g を提供するために必要な基本機能を提供するサービスである。ベーシックサービスの例としては、ユーザ管理サービス 1 1 a 、課金管理サービス 1 1 b 、セキュリティサービス 1 1 c 、 S L M (Service Level Management) サービス 1 1 d などがある。なお、 S L M とは、A S P がユーザに対してサービスの質(稼働率やレスポンス)を保証するためのシステム管理と、このシステム管理機能を使用したサービスレベル保証を 20 いう。

25 ユーティリティサービスは、A S P サービス 1 8 a ~ 1 8 g を容易に開始するためのサービスである。A S P 1 8 は、自己のサービスに対してユーティリティサービスを必要に応じて組みこむことで、A S P 1 8 自身で提供することが困難なサービスを簡単に提供可能となる。ユーティリティサービスの例としては、情報分析サービス、情報提供サービス、印刷サービスなどがある。

コミュニケーションサービスは、ユーザ 2 とA S P 1 8 との間のコミュニケーションを確立するためのサービスである。コミュニケーションサービスの例としては、ポータルサービス、ユーザ 2 の電子メールの使用を可能とする電子メールサービス 1 1 k などがある。

バックエンドサービスは、A S P サービス 1 8 a ~ 1 8 g の運用のためにバックエンドで処理を実行するサービスである。バックエンドサービスの例としては、E D I (Electronic Data Interchange) / E A I (Enterprise Application Integration) サービス、ジョブコントロールサービス 1 1 i などがある。

- 5 なお、E D I とは、商取引に関する情報を標準的な書式に統一して、企業間で電子的に交換する仕組みである。この仕組みでは、受発注や見積もり、決済、出入荷などに関わるデータを予め定められた形式にしたがって電子化し、専用線やV A Nなどのネットワークを通じて送受信する。

また、E A I とは、企業内で業務に使用される複数のコンピュータシステムを有機的に連携させ、データやプロセスの効率的に統合し、またこの統合を支援する技術及びソフトウェアである。

さらに、本実施の形態においては、A S P サービス 1 8 a ~ 1 8 g と付加サービス 1 1 a ~ 1 1 j との間の連携方法に基づいて、付加サービス 1 1 a ~ 1 1 j をユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスの3種に分類する。

ユーティリティ連携サービスは、ユーザ2又はA S P 1 8 から必要に応じて呼び出されるサービスであり、主に上記のユーティリティサービス及びコミュニケーションサービスが含まれる。ユーティリティ連携サービスの呼び出しは、ユーティリティ連携サービスにアクセスするルーチンをA S P アプリケーションの内部に組みこむことで実現する。

ベーシック連携サービスは、ユーザ2がA S P 1 8 のA S P アプリケーションにアクセスする際に介在するサービスである。ユーザ2の認証を行うユーザ管理サービス 1 1 a 、A S P 1 8 によるユーザ2に対する課金管理サービス 1 1 b などの上記ベーシックサービスは、ベーシック連携サービスに該当する。ベーシック連携サービスは、ユーザ2の操作するクライアントからA S P 1 8 に対するサービス実行リクエストが発生する度に自動的に起動される。

加えて、このベーシック連携サービスには、A S P サービス 1 8 a ~ 1 8 g から付加サービス 1 1 a ~ 1 1 j を使用する際に介在するサービスも含まれる。付加サービス業者 1 9 からA S P 1 8 に対する課金管理などのベーシックサ

サービスは、ベーシック連携サービスに該当する。ベーシック連携サービスは、ASPサービス18a～18gが付加サービス11a～11jを呼び出す際に自動的に起動される。

バックエンド連携サービスは、ASP18自体の稼動状態を監視・制御するサービスである。⁵上記バックエンドサービスは、このバックエンド連携サービスに該当する。バックエンド連携サービスは、付加サービス業者19側からASP18に対して処理の起動・停止などの操作を行う。

このようなユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスとASPサービス18a～18gとの間の連携を実現するため¹⁰に必要な3種類のインターフェースについて説明する。

各種ASPサービス18a～18g及び付加サービス11a～11jは、それぞれに対応するプログラム又はハードウェアにリクエストが出力された場合、またはそれぞれに対応するプログラム又はハードウェアが呼び出された場合に実行される。

¹⁵ユーティリティ連携サービスとASPサービス18a～18gの連携、ベーシック連携サービスとASPサービス18a～18gの連携、バックエンド連携サービスとASPサービス18a～18gとの連携は、以下のアーキテクチャで実現される。

ユーティリティ連携サービスとASPサービス18a～18gとの間の連携を可能するために、情報処理サービス提供システムは、ASPサービス18a～18gからユーティリティ連携サービスを呼び出すためのAPIを含むライブラリ8を具備する。そして、ASPサービス18a～18gを提供する時点で、ライブラリ8を使用してASPサービス18a～18g側から付加サービス11a～11jを呼び出す。これにより、ユーティリティ連携サービスとASPサービス18a～18gとの間の連携が可能となる。²⁵

なお、呼び出し側のASPサービス18a～18gと呼び出される側のユーティリティ連携サービスとが別のハードウェア上で稼動する場合があるため、本実施の形態においても、上記第1の実施の形態と同様に、ライブラリ8をネットワーク経由で使用可能とすることが望ましい。

ベースック連携サービスとA S Pサービス1 8 a～1 8 gとの間の連携を可能とするために、情報処理サービス提供システムは、ユーザ2の操作するクライアントからのリクエストを受け付けるネットワークサーバ（例えばWebサーバなどのプロセス）2 3の内部に、ベースック連携サービスの実行を依頼するユーザ

5 側リクエスト取得機能2 4を具備することで連携を可能とする。

この場合の処理の流れは、まずネットワークサーバ2 3がユーザ2からA S Pサービス1 8 a～1 8 gのリクエストを受け取る。ユーザ側リクエスト取得機能2 4のリクエスト取得機能2 4 aはこのユーザ2からA S Pサービス1 8 a～1 8 gのリクエストを取得し、依頼機能2 4 bはベースック連携サービスを提供する機能に実行を依頼する。これにより、一旦ベースック連携サービスを提供する機能に制御が移る。そして、ユーザ認証などの必要な機能を実行し、ネットワークサーバ2 3は、A S Pサービス1 8 a～1 8 gにリクエストを発する。これにより、A S P1 8 a～1 8 gを提供するアプリケーションに制御が移る。

また、情報処理サービス提供システムは、A S Pサービス1 8 a～1 8 gを提供するアプリケーションからユーティリティ連携サービスに対するリクエストをリクエスト取得機能2 5 aで取得し、依頼機能2 5 bでベースック連携サービスを提供する機能に処理実行を依頼するアプリケーション側リクエスト取得機能2 5を具備する。

バックエンド連携サービスとA S Pサービス1 8 a～1 8 gとの間の連携を可能するために、情報処理サービス提供システムは、A S Pサービス1 8 a～1 8 gを提供するアプリケーションのプロセス状態を定常的に監視し、その状態に応じてバックエンド連携サービスからA S Pサービス1 8 a～1 8 gを実現するアプリケーションのプロセスを操作し、連携を実現する。なお、A S Pサービス1 8 a～1 8 gが独立したプロセスでなく、アプリケーションサーバ上で動作するスレッドの場合には、アプリケーションサーバの持つ監視機能をバックエンド連携サービスから呼び出す。

上記のような連携のためのアーキテクチャを実現するためのリクエスト取得機能について以下で詳細に説明する。

上述したように、ユーザ側リクエスト取得機能2 4とアプリケーション側リク

エスト取得機能25とを含む2種類のリクエスト取得機能は、リクエストを取得し、このリクエストに対応するサービス又は付加サービスの提供を依頼する。

図5は、ユーザ側リクエスト取得機能24による連携状態の一例を示すブロック図である。

5 ユーザ側リクエスト取得機能24は、ユーザ2からASPサービス18a～18gに対するリクエストを取得し、ベーシック連携サービスを呼び出す。この呼び出しにより、各種ASPサービス18a～18gを使用するユーザ2が識別され、ユーザ認証及び課金処理が実行される。ASP18a～18gを提供するアプリケーションは、ユーザ2からのリクエストを受け付け、結果を返す。

10 すなわち、このユーザ側リクエスト取得機能24は、ユーザ2からのASPサービス18a～18gの使用に介在し、ユーザ認証や与信・課金などのユーザ管理機能の実行を依頼する。

図6は、アプリケーション側リクエスト取得機能25による連携状態の一例を示すブロック図である。

15 アプリケーション側リクエスト取得機能25は、ASPサービス18a～18gを提供するアプリケーションからユーティリティ連携サービスを使用する旨のリクエストを取得し、ベーシック連携サービスを呼び出す。この呼び出しにより、ASP18が付加サービス11a～11jを使用した量に応じて、ASP18に対して適正に課金することが可能になる。ユーティリティ連携サービスを提供する機能は、ASP18からのリクエストを受け付け、結果を返す。

20 すなわち、このアプリケーション側リクエスト取得機能25は、ASP18による付加サービス11a～11jの使用に介在し、認証や与信・課金などの管理処理の実行を依頼する。

以下に、リクエスト取得機能24、25の実現方法を説明する。なお、以下においてはユーザ2からWebサーバを経由してASP18にリクエストが取得される場合について説明する。しかしながら、リクエスト取得機能24、25は、ユーザ2からのアクセスを受け付けるあらゆる種類のネットワークサーバ23に同様に適用可能である。以下で説明する付加サービスの例としては、IDC21を使用する1以上のASPを使用するユーザの情報を統合して管理するサービス、

1 以上のASPの課金情報を統合して管理するサービス、1 以上のASPに関する稼動状態を一元管理するサービス、サービスを提供する1 以上の機能の稼動状態を一元管理するサービスなどがある。

リクエスト取得機能24、25は、ASPサービスを提供する機能を呼び出す
5 リクエスト又はユーティリティ連携サービスを提供する機能を呼び出すリクエストを取得し、ベーシック連携サービスを提供する機能にリクエストに関わる情報を出力する。これにより、上記の付加サービスが実行される。ユーザ2はWebサーバを経由してASPアプリケーションを使用する。

ベーシック連携サービス31には、ユーザ管理サービス11a、課金管理サービス（予信・課金サービス）11bなどがある。ユーティリティ連携サービス35には、電子メールサービス11kなどがある。

ユーザ管理サービス11aは、指定されたパスワードによりユーザを認証し、またユーザに要求されたデータへのアクセス権を確認するサービスである。

課金管理サービス11bは、指定されたユーザの予信状態（使用料の上限に達しているか否か等）を調査し、課金用の情報を保持して月毎に集計するサービスである。

電子メールサービス11kは、大量の送付先に広告用のメールを送り、その中に各送付先に特化した情報（宛先ユーザの名前など）を含めるサービスである。

図7は、リクエスト取得機能24、25を含む情報処理サービス提供システムのアーキテクチャの一例を示すブロック図である。

また、図8は、情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係の一例を示すブロック図である。

ユーザ側リクエスト取得機能24とアプリケーション側リクエスト取得機能25とは、機能的に似た点が多いため共通部分を部品化できる。共通化した部品をリクエスト取得フレームワーク26とする。

リクエスト取得フレームワーク26は例えばオブジェクト指向プログラミング言語で開発される。リクエスト取得フレームワーク26の使用する対応テーブルは、ファイル等に記録されて管理される。この対応テーブルを使用することで、リクエストがどのサービスに対応するか特定され、リクエストの種類によって起

動するベーシック連携サービスの機能を切り換え可能となる。

ユーザ側リクエスト取得機能24は、Webサーバ28又はその前段のProxyサーバ（以下、「Gateサーバ」という）27に含まれる機能とする。すなわち、Gateサーバ27は、ユーザ側リクエスト取得機能24を実現している。

ユーザ2はWebブラウザ29からWebサーバ28にアクセスする。すると、Gateサーバ27のユーザ側リクエスト取得機能24は、Webブラウザ29からWebサーバ28へのリクエストを取得する。そして、ユーザ側リクエスト取得機能24は、ベーシックサービスアダプタ30（例えばユーザ管理アダプタ30a、課金管理アダプタ30b）を用いて、ベーシック連携サービス31（例えばユーザ管理サービス31a、課金管理サービス31b）を提供する機能（例えばプロセス）に実行を依頼する。

この結果、アクセスしたユーザ2が特定され、その課金情報が蓄えられ、アクセスしたユーザ2がアプリケーションサーバ32上のASPアプリケーション33を使用可能か否か判断される。

ユーザ2にASPアプリケーション33の使用が許可されると、ユーザ2はASPアプリケーション33を使用する。さらに、このASPアプリケーション33に加えてユーザ2がユーティリティ連携サービス35（例えば電子メールサービス35a）を使用する場合、アプリケーション側リクエスト取得機能25は、ASPアプリケーション33からユーティリティ連携サービス35を提供する機能（例えばプロセス）へのリクエストを取得する。

そして、アプリケーション側リクエスト取得機能25は、ベーシックサービスアダプタ30を用いて、ベーシック連携サービス31を提供する機能に実行を依頼する。

この結果、ユーザ2が特定され、その課金情報が蓄えられ、ユーザ2がユーティリティ連携サービス35を使用可能か否か判断される。

ユーティリティ連携サービス35の使用が許可されると、ASPアプリケーション33はユーティリティサービスアダプタ34（例えば電子メールサービスアダプタ34a）を使用し、ユーティリティ連携サービス35を提供する機能に実

行を依頼する。

以上のような連携により、ユーザ管理サービス 11a、課金管理サービス 11b が ASP18 に提供される。

この結果、ASP18 は、本来サービスとして備えていない電子メールサービス 11k をユーザ 2 に提供でき、自分のサービスに新たな価値を加えることができる。

付加サービス業者 19 は、電子メールサービス 11k を ASP18 に提供した場合に、その使用に応じた課金情報を蓄え、ASP18 に請求することができる。

図 9 は、リクエスト取得機能 24、25 の構造の一例を示すクラス図である。

この図は設計表記法の一つである UML (Unified Modeling Language) の形式で記述されている。

また、図 10 は、リクエスト取得機能 24、25 の構造と具体的なサービスとの関係の一例を示すクラス図である。

矩形及び 3 分割された矩形は、クラスを表している。矩形の中に記述された名前（例えば「Gate サーバ」）はこのクラスの名前である。3 分割された矩形はクラスをより詳細に表記している。矩形の最上部の中身（例えば「取得ライブラリ」）はクラス名を、矩形の最下部の中身（例えば「前処理()」、「後処理()」）は、プロジェクト指向の場合にはメソッドに該当し、そのクラスが受け付けることのできる処理要求を示す。

破線の矢印は、根元側のクラスが先端側のクラスにアクセスして処理を呼び出したり情報を得たりすることを表している。

クラス 30（例えばユーザ管理アダプタ 30a、課金管理アダプタ 30b）は、ベーシックサービスアダプタ規約 38 に準拠し、処理メソッドを持ち、取得ライブラリ 37 から呼び出される。

クラス 30 と取得ライブラリ 37 とは、ユーザ側リクエスト取得機能 24 とアプリケーション側リクエスト取得機能 25 との共通の機能である。

取得ライブラリ 37 は、リクエスト取得機能 24、25 の基礎になるクラスである。

Gate サーバ 27 は、取得ライブラリ 37 を呼び出し、ユーザ側リクエスト

取得機能 24 を実行する。

ユーティリティサービスアダプタ 34（例えば電子メールサービスアダプタ 34a）は、ユーティリティサービスアダプタ規約 39 に準拠し、ASP アプリケーション 33 によってユーティリティ連携サービス 35（例えば電子メールサービス 35a）が呼び出される場合に使用される。

ユーティリティサービスアダプタ 34 は、取得ライブラリ 37 を呼び出す。

リクエスト取得機能 24、25 を構成する各クラスの役割を以下に示す。

Gate サーバ 27 は、クライアント 29 から ASP アプリケーション 33 に対するリクエストを仲介する Proxy サーバである。Gate サーバ 27 は、リクエストを転送する前と後の少なくとも一方で、取得ライブラリ 37 を呼び出す。

ASP アプリケーション 33 は、ASP サービスを提供する機能である。ASP アプリケーション 33 は、サービスの実行中にユーティリティ連携サービス 35（例えば電子メールサービス 35a）を使用する。

取得ライブラリ 37 は、リクエスト取得機能の基礎となる。この取得ライブラリ 37 は、ASP アプリケーション 33 又はユーティリティ連携サービス 35 を提供する機能の実行前に呼び出される前処理メソッドと、実行後に呼び出される後処理メソッド、及びアダプタに渡す情報を受け取るための情報設定メソッドを含む。

前処置メソッド及び後処理メソッドは、サービスアダプタ対応付け部 40 の内容にしたがって、ベーシックサービスアダプタ 30 を呼び出す。ベーシックサービスアダプタ 30 は、ベーシックサービスアダプタ規約 38 に準拠する。

ベーシックサービスアダプタ規約 38 は、アダプタオブジェクトが準拠すべき規約である。アダプタオブジェクトは、リクエスト取得機能 24、25 によって起動されるベーシック連携サービス 31 を提供する機能を取得ライブラリ 37 から起動する。

具体的には、ベーシックサービスアダプタ規約 38 では、処理メソッド及び情報設定メソッドを具備する旨が定義されている。この規約に準拠しているアダプタであれば、ベーシックサービスアダプタとして任意のアダプタを使用可能であ

る。iDC業者20は、ベーシックサービスアダプタ規約38に準拠しているアダプタを用意することにより、ASP18に提供するベーシックサービスを隨時変更できる。

5 ベーシックサービスアダプタ30（例えば予信・課金アダプタ、ユーザ管理アダプタ）は、ベーシックサービス規約に準拠したクラスである。

ベーシック連携サービス31（例えばユーザ管理サービス31a、課金管理サービス31b）は、ユーザ認証、課金、実行状態ログの記録などを実行するサービスである。

10 ユーティリティサービスアダプタ規約39は、ASPアプリケーション33から使用されるユーティリティサービスアダプタ34が準拠すべき規約である。

15 ユーティリティサービスアダプタ34（例えば電子メールサービスアダプタ34a）は、ユーティリティ連携サービス35を提供する機能にASPアプリケーション33からのリクエストを転送する役割を果たすクラスである。このクラスでは、転送の前後に取得ライブラリ37の前処理メソッド、後処理メソッドを呼び出す。

20 ユーティリティ連携サービス35（例えば電子メールサービス35a）は、ASPアプリケーション33から使用されるサービスである。

図11は、ユーザ側リクエスト取得機能24の前段動作の一例を示すシーケンス図である。この図11は、ユーザがASPアプリケーション33を使用するまでの処理について示している。

また、図12は、ユーザ側リクエスト取得機能24の後段動作の一例を示すシーケンス図である。この図12は、ユーザ2がASPアプリケーション33の使用を終えた後の処理について示している。

25 ユーザ2のWebブラウザ29によってASPアプリケーション33が使用される前において、ユーザ管理サービス31aを提供する機能は、ユーザ2の認証チェックを行い、課金管理サービス31bを提供する機能は、ユーザ2のASPサービス使用料のチェックを行う。

また、ユーザ2によってASPアプリケーション33が使用された後において、課金管理サービスを提供する機能は、ユーザ2の課金情報を蓄積する。

シーケンス図中の実線の矢印はメソッドの呼出を、破線の矢印は呼び出された処理を終えて制御が戻ることを示している。両矢印には、必要に応じて呼び出されるメソッド名（実線の矢印の場合、引数は（）で囲まれて記述されている）と返される値が付されている。返される値も（）で囲まれて記述されている。取得
5 ライブラリ 3 7 を実現する上で重要な働きを持つメソッドの呼び出しは、太線で強調している。

まず、図 1 1 に示すように、ユーザ 2 は、Web ブラウザ 2 9 などのクライアントを用いて ASP アプリケーション 3 3 の使用を要求する。この時点でユーザ 2 は Web サーバ 2 8 に未だアクセスしておらず、実際には Gate サーバ 2 7
10 がこのリクエストを受け付けている。

次に、Gate サーバ 2 7 は、ユーザ 2 によって指定された URL に基づいて、依頼されているサービスが何か把握する検索処理を実行する。この検索処理は、サービス - URL 対応付け部 4 1 に依頼する。

表 1 は、サービス - URL 対応付け部 4 1 が使用する対応テーブルを例示して
15 いる。

表 1

サービス名	URL
ASP トップ	/ASP/ASP.jsp
課金情報閲覧	/ASP/accounting.jsp
付帯	/ASP/option.jsp
システム連携1	/ASP/trans.jsp
システム連携2	/servlet/asp.transSvr
XYZ	/XYZ/xyz.jsp

この表 1 を参照することで、URL から 「XYZ」 などのようなサービス名を得ることができる。

次に、Gate サーバ 2 7 は、このようにして得たサービス名及びユーザ 2 が
20 入力したユーザ名などの情報を取得ライブラリ 3 7 に渡す。サービス名、ユーザ名などの情報は、後に取得ライブラリ 3 7 から各アダプタに渡され、ベーシック連携サービスを提供する機能が処理を実行するための引数として使われる。

表 2 は、ユーザ側リクエスト取得機能 2 4 がベーシック連携サービスを提供す

る機能に引数として渡す値の例である。

表 2

項目名	値
利用者名	“ABCD”
パスワード	“1111111111”
呼出側サービス名	“XYZ”
呼出先サービス名	“Gateサーバ”
データサイズ	-
データ型	-
終了状態	-

表3は、アプリケーション側リクエスト取得機能25がベーシック連携サービスを提供する機能に引数として渡す値の例である。

表 3

項目名	値
利用者名	“ASP33管理者”
パスワード	“aaaaaa”
呼出側サービス名	“電子メール. メイル送信”
呼出先サービス名	“XYZ”
データサイズ	-
データ型	-
終了状態	-

5

次に、Gateサーバ27は、取得ライブラリ37の前処理メソッドを実行する。すると、取得ライブラリ37は、サービスーアダプタ対応付け部40にアダプタ取得を依頼し、ASPアプリケーション33の前処理に必要なアダプタ群を取得する。

10 表4は、サービスーアダプタ対応付け部40が使用するテーブルを例示している。

具体的には、取得ライブラリ37は、既に表2のような情報を受け取っているため、「XYZ」をキーワードとして表4に対する検索を行う。

これにより、取得ライブラリ37は、ASPアプリケーション33の前処理に
15 必要なアダプタとしてユーザ管理アダプタ30a、課金管理アダプタ30bを選択する。

表 4

サービス	アダプタ	
XYZ. 前処理	ユーザ管理、課金管理	ユーザ側リクエスト 取得機能用
XYZ. 後処理	課金管理	
システム連携1. 前処理	ユーザ管理	
システム連携2. 後処理	課金管理、 付加サービス利用用	
...	...	
電子メール. メール送信. 前処理	ユーザ管理、課金管理	アプリケーション側 リクエスト 取得機能用
電子メール. メール送信. 後処理	課金管理	
電子メール. メール受信. 前処理	ユーザ管理、課金管理	
電子メール. メール受信. 後処理	課金管理、 付加サービス利用用	
...	...	

次に、取得ライブラリ 3 7 は、選択したアダプタを使用して実際のサービスを提供する機能を呼び出す。取得ライブラリ 3 7 は、先にユーザ管理サービス 3 1 a を提供する機能を呼び出し、次に課金管理サービス 3 1 b を提供する機能を呼び出す。具体的には、取得ライブラリ 3 7 は、先の処理で設定された表 2 の情報を選択されたアダプタに渡し、各アダプタは、自己の備えている処理メソッドを呼び出す。

ユーザ管理アダプタ 3 0 a は、ユーザ管理サービス 3 1 a を提供する機能に特化した操作メソッドを呼び出すとともに、ユーザ管理サービス 3 1 a を提供する機能にユーザ名とパスワードとを渡す。

すると、ユーザ管理アダプタ 3 0 a は、ユーザが正当であるか否かチェックした結果を、ユーザ管理サービス 3 1 a を提供する機能から受け取る。その結果は、さらにユーザ管理アダプタ 3 0 a から取得ライブラリ 3 7 に返される。

チェック結果に問題がなければ、同様に課金管理アダプタ 3 0 b は課金管理サービス 3 1 b のメソッドを呼び出す。これにより、課金管理サービス 3 1 b は、例えばこのユーザの使用料が契約上の上限値を上回っているか否かチェックする予信管理を行う。

この処理が終わると、制御は取得ライブラリ 3 7 から Gate サーバ 2 7 に戻る。

Gate サーバ 2 7 は、Web サーバ 2 8 に対してユーザ 2 が指定した URL

を通知し、A S P アプリケーション3 3によるサービスが提供される。サービスとしては、例えば発注システムの機能提供、スケジュール管理ツールの機能提供などがある。

続いて、図1 2に示すように、Gateサーバ2 7は、情報設定メソッドを用いて、ベーシック連携サービスを提供する機能に渡され後処理で使用される情報を、取得ライブラリ3 7に引き渡す。引き渡される情報には、例えば課金を行うためのユーザの接続時間やログを記録するための時刻情報などがある。

次に、Gateサーバ2 7は、取得ライブラリ3 7の後処理メソッドを呼び出す。すると、前処理の場合と同様に、取得ライブラリ3 7は、サービスーアダプタ対応付け部4 0から必要なアダプタを取得し、該当するアダプタの処理メソッドを呼び出す。これにより、ベーシック連携サービス（この例では課金管理サービス3 1 b）が提供される。

図1 3は、アプリケーション側リクエスト取得機能2 5の前段動作の一例を示すシーケンス図である。この図1 3は、ユーザ2が電子メールサービス3 5 aを使用するまでの処理について示している。このようにサービスーアダプタ対応付け部4 0を用いることで、A S P毎に、提供される付加サービスの種類を切り換えることができる。また、表4のようにテーブルで設定すれば、このテーブルを更新するだけでA S Pに提供される付加サービスの種類を変更できるためi D C業者2 0は管理・運用が容易になり、A S Pに提供する付加サービスの種類に柔軟性を持たせることができる。

また、図1 4は、アプリケーション側リクエスト取得機能2 5の後段動作の一例を示すシーケンス図である。この図1 4は、ユーザ2が電子メールサービス3 5 aの使用を終えた後の処理について示している。

この図1 3及び図1 4は、すでにA S Pアプリケーション3 3が使われている時点から記述されている。また、図1 3及び図1 4は、ユーザ2からのメール送信のリクエストに応じるために電子メールサービス3 5 aが使用される場合を例示している。

A S Pアプリケーション3 3の処理の一部によってユーティリティ連携サービスの1つである電子メールサービス3 5 aが使用される前に、ユーザ管理サービ

ス31aの依頼側であるASP18の認証チェックが行われ、課金管理サービス31bにより処理依頼側であるASP18の付加サービス使用料がチェックされる。

また、ASPアプリケーション33の処理の一部によってユーティリティ連携サービスの1つである電子メールサービス35aが使用された後に、課金管理サービス31bにより依頼側であるASP18の課金情報が蓄積される。

まず、図13に示すように、ASPアプリケーション33は電子メールサービスアダプタ34aを生成し、アダプタ34aに対してメール送信を要求する。

すると、電子メールサービスアダプタ34aは、取得ライブラリ37に対して情報渡す。

取得ライブラリ37は、前処理メソッドを呼び出し、実際の電子メールサービス35aを提供する機能に対してメール送信処理を依頼し、図14に示すように後処理メソッドを呼び出す。

取得ライブラリ37は、アダプタを取得するためのキーワードとしてサービス名「電子メール」に加え、使用するメソッド名「メール送信」も使用する。その他の取得ライブラリ37の動作は、ユーザ側リクエスト取得機能24の場合と同様である。ここでも、ユーザ側リクエスト取得機能24と同様に表4のようなサービスアダプタ対応付けテーブルを使用する。これにより、ASP18に提供される付加サービスの種類が固定されず、より柔軟性を持たせることができるため、iDC業者20はASP18の要望にあった付加サービスを提供できる。また、ASP18との契約内容に変更が生じてもこのテーブルを更新するだけで継続使用可能であり、ライブラリ自体の再コンパイルが不要であるため、iDC業者20側の運用コストを軽減させることができる。

なお、必要になった時点でそれぞれのアダプタを初期化する場合、アダプタの選択要求からサービス提供までに処理時間がかかる。したがって、事前にアダプタ群を生成してキャッシング（保留）しておき、処理を高速化してもよい。

以上のような情報処理サービス提供システムを実現すると、iDC業者20、ASP18は以下のようない点を得ることができる。

iDC業者20は、iDC21上にアプリケーションプラットフォーム22を

具備することにより、単に iDC21 を使用するのみでは実現できない各種付加サービスを提供できる。これにより、iDC 業者20は、異なる機能の使用を求めるそれぞれのASPに対して適切な機能を容易に提供できる。また、iDC 業者20は、付加サービスを使用したユーザ2又はASP18からその使用に応じた対価を得ることができる。

iDC 業者20は、ASP18との契約内容に変更が生じた場合にその変更内容を容易に事業に反映させることができ、提供するサービスを実現する機能の種類、サービスの内容を容易に追加／変更できる。これにより、iDC 業者20は、事業の運用・保守コストを削減でき、利益効率を高めることができる。

iDC 業者20は、ASP18に対して他のASPの提供する付加サービスを提供することで、データセンタ単独での事業を行う場合よりも自データセンタの付加価値を高めることができ、自データセンタを使用するユーザ2、ASPを増加させることができる。

ASP18は、自己の提供するASPサービスに付加サービスを連携させることができる。

また、ASP18は、他のASPと連携してユーザ2にサービスを提供し、ユーザ2の使用可能な機能を増加させてユーザ2の利便性を増すことができ、ユーザ2に使用される機会を増加させることができる。

また、ASP18は、付加サービスを提供するために必要な技術力を持たない場合であっても、他のASPによって提供される付加サービスを使用することで、容易にサービスの充実を図ることができる。

また、ASP18は、本来ASPサービスの提供用として作られていないアプリケーションをASPサービス用として機能させる場合に、ASPとして機能するために必要な機能（課金管理やシステム監視などの機能）を付加サービスで補充できる。したがって、ソフトウェアのASP化が容易になる。

また、ASP18は、他のASPとの間でユーザ管理や課金管理を共通化できるため、ユーザ2に対して複数のASPに対するシングルサインオン環境を提供でき、課金請求を統合できる。複数のASPに対するシングルサインオン環境や課金請求の統合は、ユーザ2にとっても管理が容易となるため、ユーザの獲得を

促進できる。

(第3の実施の形態)

上記各実施の形態に係る情報処理サービス提供システムにおける各機能、各要素は、同様の作用・動作を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせてもよく、各機能、各要素を分割してもよい。

図15は、本実施の形態に係る情報処理サービス提供システムの構成を例示するブロック図である。この図15は、上記図7に示す情報処理サービス提供システムの変形例を示す。

ベーシック連携サービス31a～31cを提供する機能は、ベーシック連携アダプタ31a～31cを経由してリクエストを受け又は発する。

ユーティリティ連携サービス35を提供する機能は、ユーティリティサービスアダプタ34を経由してリクエストを受け又は発する。

リクエスト取得サーバ42は、リクエスト取得機能42a、42bとサービスアダプタ対応付け部40を具備する。

リクエスト取得サーバ42は、Webブラウザ29、付加サービス31a～31c、35を提供する機能、ASPアプリケーション33のそれぞれからリクエストを取得し、サービスアダプタ対応付け部40の内容にしたがって取得したリクエストに応じて、ASPアプリケーション33又は付加サービス31a～31c、35を提供する機能を選択し、選択したASPアプリケーション33又は付加サービス31a～31c、35を提供する機能にリクエストを送る。

これにより、あらゆる情報処理サービスが他の情報処理サービスを使用できる関係となり、サービス間の連携が効率的に実現される。その結果として、ASPアプリケーションと複数の付加的機能を組み合わせた機能の構築やカスタマイズが容易となる。

例えば付加サービスを提供する機能から他の付加サービスを提供する機能に実行を依頼することができる。そして、サービスを提供する複数の機能を自由に連携させて一つの情報処理サービスを提供できる。

上記各実施の形態に係る情報処理サービス提供システムの各機能、各要素は、コンピュータに実行させることのできるプログラム、モジュール、プロセスとし

て実現される。このプログラムにより上記各機能、各要素を提供する場合には、
例えは磁気ディスク（フレキシブルディスク、ハードディスク等）、光ディスク
（CD-ROM、DVD等）、半導体メモリなどの記録媒体¹⁶にそのプログラ
ムを書き込んでコンピュータに適用可能である。また、このプログラムは通信媒
5 体によりコンピュータ、コンピュータシステムに伝送してよい。コンピュータは、
プログラムを読み込み、そのプログラムによって動作が制御されることにより、
上述した処理を実行する。

While the description above refers to particular embodiments of
the present invention, it will be understood that many modifications may
10 be made without departing from the spirit thereof. The accompanying
claims are intended to cover such modifications as would fall within the
true scope and spirit of the present invention. The presently disclosed
embodiments are therefore to be considered in all respects as
illustrative and not restrictive, the scope of the invention being
15 indicated by the appended claims, rather than the foregoing description,
and all changes that come within the meaning and range of equivalency of
the claims are therefore intended to be embraced therein. For example,
the present invention can be practiced as a computer readable recording
medium in which a program for allowing the computer to function as
20 predetermined means, allowing the computer to realize a predetermined
function, or allowing the computer to conduct predetermined means.